UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/702,320	10/31/2000	Laurence R. Simar, Jr.	TI-30559	9784

23494 7590 02/19/2004

TEXAS INSTRUMENTS INCORPORATED
P O BOX 655474, M/S 3999
DALLAS, TX 75265

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 02/19/2004

10

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/702,320	SIMAR, JR. ET AL.
	Examiner	Art Unit
	David J. Huisman	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 January 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4 and 7-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4 and 7-11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 January 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-2, 4, and 7-11 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: #8. Extension of Time (1 month) as received on 1/13/2004 and #9. Amendment "A" as received on 1/13/2004.

Specification

3. The abstract of the disclosure is objected to because of the following minor informalities: In line 2 of the abstract, replace "in" with --an-- and replace "includes" with either --including--, --that includes--, or --which includes--. Correction is required. See MPEP § 608.01(b).

4. The disclosure is objected to because of the following informalities: In the amendment made to the paragraph at page 7, line 16, to page 8, line 5, replace the phrase "external peripherals 80 via bus 80" with --external peripherals 82 via bus 80--.

Appropriate correction is required.

Claim Objections

5. Claim 1 is objected to because of the following informalities: Please insert the word --and-- before "adding" in line 24 on page 12. Appropriate correction is required.

6. Claim 1 is objected to because of the following informalities: Please replace "execution packet" with --execute packet--. Appropriate correction is required.

7. Claim 4 is objected to because of the following informalities: Insert a comma before "or" in line 16 of the claim. Appropriate correction is required.

8. Claim 4 is objected to because of the following informalities: For increased readability, the examiner recommends inserting a comma before "dependent" in line 17 of the claim. Appropriate correction is required.

9. Claim 4 is objected to because of the following informalities: Insert a comma before "and" in line 19 of the claim. Appropriate correction is required.

10. Claim 4 is objected to because of the following informalities: Insert the word --to-- after "according" in line 23 of the claim. Appropriate correction is required.

11. Claim 9 is objected to because of the following informalities: Insert a comma before "or" in line 7 of the claim, and before "dependent" in line 8 of the claim. Appropriate correction is required.

12. Claim 10 is objected to because of the following informalities: Please insert the word --and-- before "adding" in line 18 of the claim. Appropriate correction is required.

13. Claim 10 is objected to because of the following informalities: For increased readability, the examiner recommends inserting a comma before "not" in line 20 of the claim. Appropriate correction is required.

14. Claim 10 is objected to because of the following informalities: In line 23 of the claim, please replace "dispatching" with --dispatch--. Appropriate correction is required.

15. Claim 11 is objected to because of the following informalities: In line 11 of the claim, please insert --an-- before "execute packet". Appropriate correction is required.

16. Claim 11 is objected to because of the following informalities: Line 8 of the claim should be worded more appropriately. The examiner is not sure if the word --of-- should be inserted after "a first plurality" or if "a first plurality" should be deleted. Appropriate correction is required.

17. Claim 11 is objected to because of the following informalities: For increased readability, the examiner recommends inserting a comma before "not" in line 14 of the claim. Appropriate correction is required.

Withdrawn Rejections

18. Through amendment, applicant has overcome the rejections set forth in the previous Office Action, mailed on September 9, 2003, for claims 1-7. Therefore, those rejections are hereby withdrawn by the examiner. However, upon further consideration, a new ground(s) of rejection is made below.

Claim Rejections - 35 USC § 102

19. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

20. Claims 1-2, 7-8, and 10-11 are rejected under 35 U.S.C. 102(b) as being anticipated by Simar et al., European Patent Application, EP 0855648A2 (herein referred to as Simar).

21. Referring to claim 1, Simar has taught a digital processing system having a microprocessor (Fig.1, component 11), wherein the microprocessor comprises:

a) fetch circuitry for fetching instruction fetch packets from sequential memory address locations, wherein each fetch packet contains a first plurality of instructions, each instruction including an instruction type and a predetermined p-bit, said p-bit having a first digital state indicating a next instruction is to execute in parallel with said instruction and a second digital state indicating a next instruction is to execute in a cycle after said instruction. See Fig.3 and page 3, lines 19-30.

b) a second plurality of functional units, each of the second plurality of functional units operable to execute a corresponding instruction in parallel with other functional units. See Fig.20, Fig.21, and page 4, lines 14-18. Note that execution units L2, S2, M2, and D2, for instance, are considered a second plurality of functional units which will execute L2-type, S2-type, M2-type, and D2-type instructions, respectively. And, these units will execute instructions in parallel with the L1, S1, M1, and D1, for instance, are considered other functional units which will execute L1-type, S1-type, M1-type, and D1-type instructions, respectively.

c) dispatch circuitry (Fig.20, component 10b) connected to said fetch circuitry and said second plurality of functional units operable to:

c1) select an execution packet from one or more fetch packets, wherein an execute packet varies in size and contains only a set of instructions that can be executed in parallel on the plurality of functional units, by scanning instructions from lower memory address locations to higher memory address locations and adding an instruction to said execute packet when said p-bit of a prior instruction has said first digital state until said p-bit of

an instruction has said second digital state. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits. See Fig.8-19. For example, in Fig.14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own packet).

c2) dispatch each instruction of said selected execute packet to a functional unit corresponding to said instruction type of said instruction. See page 3, lines 45-47. Note that instructions A and B are executed in parallel, and for them to be executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an ADD instruction and instruction B is a multiply instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

22. Referring to claim 2, Simar has taught a system as described in claim 1. Simar has further taught that the first plurality is equal in number to the second plurality. For example,

Fig.7, shows a fetch packet with a first plurality of instructions equal to 4 (A, B, C, and D). Since instruction A, B, and C, have p-bits equal to 1, then all 4 instructions can be placed in a single execute packet. This means that all 4 instructions are to be executed in parallel, which would in turn require 4 execution units (L2, S2, M2, and D2). Therefore, in this case, the first plurality equals the second plurality.

23. Referring to claim 7, Simar has taught a method of operating a digital system having a microprocessor (Fig.1, component 11), wherein the microprocessor has a plurality of functional units (Fig.24) for executing instructions in parallel, comprising the steps of:

a) storing instructions at sequential memory address locations, each instruction including an instruction type and a predetermined p-bit, said p-bit having a first digital state indicating a next instruction is to execute in parallel with said instruction and a second digital state indicating a next instruction is to execute in a cycle after said instruction. See Fig.3 and note that instructions A, B, C, and D, were stored at sequential memory locations x00, x01, x10, and x11. Also, note that in addition to each instruction inherently including an instruction type, each instruction also includes a p-bit with the aforementioned digital states. See Fig.3 and page 3, lines 19-30.

b) fetching a sequence of instruction fetch packets, wherein each fetch packet contains a first plurality of instructions. See Fig.24, packets 1710, 1720, 1730, and 1740. Note that the system will fetch a sequence of instruction packets that contain a first plurality of instructions (in this case the first plurality = 8).

c) scanning the p-bit of each instruction of each fetch packet from lowest memory address location to highest memory address location to determine an execute packet dependent on the p-bits. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch

packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits. See Fig.8-19. For example, in Fig.14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own packet).

d) dispatching each instruction within the determined execute packet to one of a second plurality of execution units dependent upon an instruction type of the instruction. See page 3, lines 45-47. Note that instructions A and B are executed in parallel, and for them to be executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an addition-type instruction and instruction B is a multiply-type instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

24. Referring to claim 8, Simar has taught a method as described in claim 7. Simar has further taught that the first plurality of instructions in a fetch packet equals the second plurality of functional units. See page 5, line 54. Note that a fetch packet comprises a first plurality of 8 instructions. And, from Fig.24, it should be realized that a second plurality of 8 functional units

exist (L1, S1, M1, D1, L2, S2, M2, and D2). This is necessary when all of the instructions in the fetch packet are to be executed in parallel (they form a single fetch packet). Since each instruction would require a unique functional unit and there are 8 instructions, then 8 functional units must exist. See page 6, lines 1-3.

25. Referring to claim 10, Simar has taught a digital processing system having a microprocessor (Fig.1, component 11), wherein the microprocessor comprises:

a) fetch circuitry for fetching instruction fetch packets from sequential memory address locations, wherein each fetch packet contains a first plurality of instructions, each instruction including an indication of a corresponding functional unit and an indication of an execute packet. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 3. It should also be realized that each instruction indicates a corresponding functional unit just by the type of instruction that it is. For instance, a multiply-type instruction will inherently correspond to a multiplier execution unit (Fig.24). It will not correspond to a load/store unit, for instance, because it is not a load/store-type instruction. In addition, the p-bit of each instruction indicates an execution packet.

b) a second plurality of functional units, each of the second plurality of functional units operable to execute a corresponding instruction in parallel with other functional units. See Fig.20, Fig.21, and page 4, lines 14-18. Note that execution units L2, S2, M2, and D2, for instance, are considered a second plurality of functional units which will execute L2-type, S2-type, M2-type, and D2-type instructions, respectively. And, these units will execute instructions in parallel with the L1, S1, M1, and D1, for instance, are considered other functional units which will execute L1-type, S1-type, M1-type, and D1-type instructions, respectively.

c) dispatch circuitry (Fig.20, component 10b) connected to said fetch circuitry and said second plurality of functional units operable to:

c1) select an execute packet from one or more fetch packets, wherein an execute packet varies in size and contains only a set of instructions that can be executed in parallel on the plurality of functional units, by scanning instructions from lower memory address locations to higher memory address locations and adding an instruction to said execute packet dependent upon said indication of an execute packet, wherein upon a branch into the middle of an execute packet, not selecting instructions having memory address locations lower than the branch. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits. See Fig.8-19. For example, in Fig.14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own packet). It should be noted that Simar has taught creating an execute packet from one fetch packet. The applicant has claimed "selecting an execute packet from one **or** more fetch packets..." The use of the word "or" allows a reference to read on this limitation if

it teaches selecting an execute packet from one fetch packet **or** selecting an execute packet from more than one fetch packet. It is not required that the reference teach both. Consequently, Simar anticipates this limitation. Finally, see page 6, lines 35-39, and note that upon branching to a middle of an execute packet, all instructions at lower addresses are ignored.

c2) dispatch each instruction of said selected execute packet to a corresponding functional unit. See page 3, lines 45-47. Note that instructions A and B are executed in parallel, and for them to be executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an ADD instruction and instruction B is a multiply instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

26. Referring to claim 11, Simar has taught a method of operating a digital system having a microprocessor (Fig.1, component 11), wherein the microprocessor has a plurality of functional units (Fig.24) for executing instructions in parallel, comprising the steps of:

a) storing instructions at sequential memory address locations, each instruction including an indication of a corresponding functional unit and an indication of an execute packet. See Fig.3 and note that instructions A, B, C, and D, were stored at sequential memory locations x00, x01, x10, and x11. It should also be realized that each instruction indicates a corresponding functional unit just by the type of instruction that it is. For instance, a multiply-type instruction will inherently correspond to a multiplier execution unit (Fig.24). It will not correspond to a

load/store unit, for instance, because it is not a load/store-type instruction. In addition, the p-bit of each instruction indicates an execution packet. See page 3, lines 19-30, and page 5, line 54, to page 6, line 3.

b) fetching a sequence of a first plurality of instruction fetch packets, wherein each fetch packet contains a first plurality of instructions. See Fig.24, and note that the system has fetched packets 1710, 1720, 1730, and 1740. Note that the system will fetch a sequence of instruction packets that contain a first plurality of instructions (in this case the first plurality = 8).

c) scanning the indication of an execute packet of each instruction of each fetch packet from lowest memory address location to highest memory address location to determine an execute packet, wherein upon a branch into the middle of an execute packet, not selecting instructions having memory address locations lower than the branch. See Fig.3, page 3, lines 19-30, and page 5, line 54, to page 6, line 7. Note that the fetch packet comprises 4 instructions (A, B, C, and D) and that different size execution packets are formed based on the values of the p-bits (indication of an execute packet). See Fig.8-19. For example, in Fig. 14, instructions are read from lower memory address to higher memory address (as shown in Fig.3), starting with instruction A. Instruction A is added to the execute packet and instruction A's p-bit is checked and determined to be 1. Therefore, this p-bit value indicates that the next instruction (instruction B), may execute in parallel with instruction A. As a result, instruction B is added to the execute packet. Instruction B's p-bit is 0, which means that instruction C must start a new execute packet since it may not execute in parallel with instruction B. The final execute packets are then shown in Fig.15 (note that instructions A and B are in the same packet and C is in its own

packet). Finally, see page 6, lines 35-39, and note that upon branching to a middle of an execute packet, all instructions at lower addresses are ignored.

d) dispatching each instruction within the determined execute packet to a corresponding execution unit. See page 3, lines 45-47. Note that instructions A and B are executed in parallel, and for them to be executed they must be dispatched to the execution (functional) units in a manner similar to that shown in Fig.24. For instance, if instruction A is an addition-type instruction and instruction B is a multiply-type instruction, then instruction A will be dispatched to the L1 execution unit, since the L1 unit is an arithmetic logic unit, and instruction B will be dispatched to the M1 execution unit since M1 is a multiplier unit (page 4, lines 14-16).

Claim Rejections - 35 USC § 103

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

28. Claims 4 and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Simar, as applied above, and further in view of Heishi et al., U.S. patent No. 6,324,639 (herein referred to as Heishi).

29. Referring to claim 4, Simar has taught a system as described in claim 2. Simar has not explicitly taught the specifics of claim 4. However, Heishi has taught that the dispatch circuitry comprises:

a) a first latch to hold said first plurality of instructions of a first packet. See Fig.8, component 221.

b) a second latch to hold said first plurality of instructions of a second fetch packet immediately following said first fetch packet. See Fig.8, component 222.

c) a first plurality of multiplexers (Fig.8, components 224), each multiplexer having a first input receiving an instruction from a predetermined position of said first latch (for instance, mux 224a receives an instruction from slot A0), a second input receiving an instruction from a corresponding position of said second latch (mux 224a receives an instruction from slot B0), a control input (column 13, lines 13-18, and note that component 223 in Fig.8 controls the muxes since the muxes must inherently be controlled) and an output (note the muxes output to an instruction register), each multiplexer, selecting at said output said instruction from said first latch, said instruction from said second latch, or no instruction, dependent upon said control input (see fig.9A-9F, and note that at times instructions exist within the latches but are not selected by the muxes, and sometimes instructions are selected from the first or second latches).

d) a dispatch control circuit connected to said first latch, said second latch, and said plurality of multiplexers, said dispatch control circuit receiving said predetermined p-bit from each instruction of said first latch and each instruction of said second latch for control of said plurality of multiplexers via said control inputs according to the execute packets determined by said p-bits. See Fig.11 and note that the issuing control circuitry 31 is connected to the instruction register which in turn is coupled to the multiplexers. Component 31 actually receives a p-bit from each instruction, which is the most significant bit of each instruction. See Fig.11, Fig.6A-

F, and column 9, line 50, to column 10, line 3. Basically, these bits determine which instructions are sent by the multiplexers to be decoded and issued. Therefore, they control the multiplexers.

e) a cross point circuitry connected to said plurality of multiplexers for dispatching said instructions at said output of said multiplexers to a functional unit corresponding to said instruction type of each instruction. See Fig.11, and note that the decoders are considered a cross point circuitry since after the instructions are decoded, they are dispatched to the execution units 41.

Heishi has taught that such circuitry allows for accommodation of variable length instructions. See Fig.15 and Fig.16, and note that words from multiple slots in the latches may be combined to form a larger instruction, such as one of the format shown in Fig.6D-F. A person of ordinary skill in the art would've recognized that larger size opcodes, displacement values, and immediate values may be implemented when implementing variable length instructions, thereby increasing the user's flexibility in that the user would not be restricted to such small numbers. For instance, see Fig.6B and 6E. Note that the user would be allowed to have a much larger immediate value. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Simar to include such circuitry so that more flexibility can be achieved.

30. Referring to claim 9, Simar has taught a method as described in claim 7. Simar has further taught that said step of determining an execute packet boundary dependent upon the p-bits includes:

a) storing a first fetch packet and storing a second fetch packet. See Fig.24, and note that multiple fetch packets are brought within the system for processing. Simar has not explicitly taught that the first fetch packet and the second fetch packet are actually stored in a first and

second latch, respectively. However, Heishi has taught the concept of having two latches and storing two separate fetch packets in them. See Fig.8, components 221 and 222. Also, see Fig.9B and 9C and note that a first packet comprising units 1, 2, and 3 are stored in the first latch, and units 4, 5, and 6 are stored in a second latch. A person of ordinary skill in the art would have recognized that, in general, it is more efficient to store fetch packets internally (in a latch, cache, etc.) so that a slow main memory access is not required. Having these latches will also mask the main memory access time. For instance, when one packet is being dispatched, no other packet can be dispatched. Therefore, a next packet can be retrieved from memory as opposed to just waiting until the dispatch finishes and then performing the main memory access. By overlapping (pipelining) the main memory access and dispatching, the system becomes more efficient in that when the dispatch does finish, instead of going to slow memory to get the next packet, the packet is retrieved from within the system which is much faster. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Simar in view of Heishi such that first and second fetch packets are store in first and second latches, respectively.

b) Simar in view of Heishi has further taught selecting an instruction from said first latch, a corresponding instruction from said second latch or no instruction, dependent upon said p-bit from each instruction of said first latch and each instruction of said second latch. See Fig.24. for instance and note that instructions are selected from the fetch packet, which would be stored in a latch, according to Heishi. This would also hold true for the second fetch packet (for instance, packet 1720 shown in Fig.24). And, if a p-bit indicates that a next instruction may not be executed in parallel, then that instruction will not be selected.

Conclusion

31. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (703) 305-7811. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
February 6, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100